MERIt: Meta-Path Guided Contrastive Learning for Logical Reasoning

Fangkai Jiao†, Yangyang Guo§, Xuemeng Song†, Liqiang Nie†
†School of Computer Science and Technology, Shandong University, Qingdao, China
§National University of Singapore
jiaofangkai@hotmail.com {guoyang.eric, sxmustc, nieliqiang}@gmail.com

Abstract

Logical reasoning is of vital importance to natural language understanding. Previous studies either employ graph-based models to incorporate prior knowledge about logical relations, or introduce symbolic logic into neural models through data augmentation. However, these methods heavily depend on annotated training data, and thus suffer from over-fitting and poor generalization due to the dataset sparsity. In this paper, we propose MERIt, a MEta-path guided contrastive learning method for logical ReasonIng of text, to capture relevant knowledge from the abundant unlabeled text data. Two novel strategies serve as indispensable components of our method. In particular, a strategy based on meta-path is devised to discover the logical structure in natural texts, followed by a counterfactual data augmentation strategy to eliminate the information shortcut induced by pre-training. The experimental results on two challenging logical reasoning benchmarks, ReClor and LogiQA, demonstrate that our method outperforms the SOTA baselines with significant improvements.

1 Introduction

Logical reasoning has long been recognized as one key critical thinking ability of human being. Until very recently, some pioneer researchers have crystallized this for the NLP community, and built several public challenging benchmarks, such as ReColor (Yu et al., 2020) and LogiQA (Liu et al., 2020). Logical reasoning[1] requires to correctly infer the semantic relations with respect to the constituents among different sentences. A typical formulation of logical reasoning is illustrated in Figure 1, a real-world examination (such as GMAT) instance provided by ReClor. As can be seen, to find the correct answer for the given question, we need to conduct the logical reasoning by extracting the logical structures residing in each context-option pair, and justifying its reasonableness.

Context: Economist: **(1)** A country's rapid emergence from an economic recession (r_1) requires **(2)** substantial new investment in that country's economy. Since **(3)** people's confidence in the economic policies of their country (r_2) is a precondition for **(2)** any new investment, **(4)** countries that put collective goals before individuals' goals (r_3) cannot **(1)** emerge quickly from an economic recession.

Question:

Which one of the following, if assumed, enables the economist's conclusion to be properly drawn?

Options:

A. People in **(4)** countries that put collective goals before individuals' goals (r_4) lack **(3)** confidence in the economic policies of their countries.

B. A country's economic policies are the most significant factor determining whether that country's economy will experience a recession.

C. If the people in a country that puts individuals' goals first are willing to make new investments in their country's economy, their country will emerge quickly from an economic recession.

D. No new investment occurs in any country that does not emerge quickly from an economic recession.

Answer: A

Logic Structure: (4) $\xrightarrow{r_4}$ (3) $\xrightarrow{r_2}$ (2) $\xrightarrow{\bar{r}_1}$ (1) $\Leftrightarrow$ (4) $\xrightarrow{r_3}$ (1)

Figure 1: An instance of logical reasoning from the ReClor dataset. To infer the right answer, we should uncover the underlying logical structure, as shown in the bottom. (x) represents the logical variable (e.g., entity or phrase) and r_j represents the relation (e.g., predicate) between the logical variables. $\bar{r}_j$ is the passive relation of r_j.

As a matter of fact, logical reasoning is still at its initial stage, thence, existing studies are relatively sparse in literature. Some efforts have been devoted to designing specific model architectures or integrating symbolic logic as the hints attached to the potential logical structure. For instance, Huang et al. (2021) and Ouyang et al. (2021) first constructed a graph of different constituents and then performed implicit reasoning with graph neural networks (GNNs). Wang et al. (2021) proposed LReasoner, a unified context extension and data augmentation framework based on the parsed logi-

[1]We refer the term *logical reasoning* to the task itself in the remaining of this paper.

cal expressions.

These approaches have achieved some progress on benchmark datasets. However, though equipped with pre-trained language models, they still suffer from problems like overfitting and poor generalization. In fact, building a model conscious of the latent logical structures beneath natural language is non-trivial, not only due to the high sparsity of the existing datasets, but also the goal of general pre-training, i.e., masked language modeling (Devlin et al., 2019), which is far from logical reasoning. To tackle this issue, we aim to devise a logical reasoning-oriented pre-training method in this work.

Our proposed method is inspired by the recent progress of contrastive learning based pre-training. It mainly consists of two novel components: meta-path guided data construction and counterfactual data augmentation. Both components are leveraged to perform automatic instance construction from unlabeled corpus (e.g., Wikipedia) in contrastive learning. Regarding the first component, we propose to employ the meta-path to define a symbolic form of logical structure. The intuition behind this is that the logical consistency can be expressed a series of relation triplets, and a meta-path inherently offers a bridge of consistency. Specifically, given an arbitrary document and a pair of entities contained in it, we endeavor to find a positive instance pair in the document according to the logical structure. And the negative ones can thus be generated by modifying the relations involved in the structure to break its logical consistency. Nevertheless, the contrastive learning often fails when models can easily locate trivial solutions. In our context, the pre-trained language model may exclude the negative options through their conflicts with the world knowledge. To eliminate this information shortcut, in our second novel component, we devise a strong counterfactual data augmentation (Zeng et al., 2020b) strategy. By mixing counterfactual data during pre-training, of which the positive instance pair is also against the world knowledge, this component shows more advantage in reasoning over logical relations.

In this paper, we integrate our method with both ALBERT (Lan et al., 2020) and RoBERTa[2] (Liu et al., 2019) for further pre-training, and then fine-tune them on two downstream logical reasoning benchmarks, ReClor and LogiQA. The experimental results demonstrate that our method can outperform all the existing strong baselines, yet without any augmentation from the original training data. Besides, the ablation studies also show the effectiveness of the meta-path in data construction and counterfactual data augmentation strategy. In short, the contribution of this paper is summarized as follows:

1. We propose MERIt, a MEta-path guided contrastive learning method for logical ReasonIng, to reduce the heavy reliance on annotated data. To the best of our knowledge, we are the first to explore self-supervised pre-training for logical reasoning.

2. We successfully employ the meta-path-based strategy to mine the potential logical structure in raw text. It is enabled automatically generate negative candidates for contrastive learning via editing on logical relations.

3. We propose a simple yet effective counterfactual data augmentation method to eliminate the information shortcut during pre-training.

4. We evaluate our method on two logical reasoning tasks, LogiQA and ReClor. The experimental results show that our method achieves the new state-of-the-art performance on both datasets.

2 Related Work

2.1 Self-Supervised Pre-training

With the success of language modeling based pre-training (Devlin et al., 2019; Yang et al., 2019; Brown et al., 2020), designing self-supervised pretext tasks to facilitate specific downstream ones has been extensively studied thus far. For example, Guu et al. (2020) proposed to train the retriever jointly with the encoder via retrieval enhanced masked language modeling for open-domain question answering. Deng et al. (2021) proposed ReasonBERT to facilitate complex reasoning over multiple and hybird contexts. The model is pre-trained on automatically constructed query-evidence pairs across different types of corpus and long-range relations. Besides, generative pre-training such as PEGASUS (Zhang et al., 2020) recovers the masked gap-sentences from the rest of the document, achieving state-of-the-art results on 12 downstream datasets for abstractive summarization.

[2]In this paper, we use *ALBERT* and *RoBERTa* to refer to ALBERT-xxLarge and RoBERTa-Large for simplicity, respectively.

In addition, contrastive learning (Hadsell et al., 2006) contributes to a strong toolkit to implement self-supervised pre-training. The key to contrastive learning is to build efficacious positive and negative counterparts. For example, Gao et al. (2021) leveraged Dropout (Srivastava et al., 2014) to build positive pairs from the same sentence while keeping the semantics untouched. Other sentences in the same mini-batch serve as negative candidates to obtain better sentence embeddings. ERICA (Qin et al., 2021) promotes the understanding of entities and their relations in text via discrimination based pretext task, where the negative candidates are sampled from the pre-defined dictionaries. Despite these successful data construction procedures, when it comes to logical reasoning, the automatic instance pair construction can be difficult. We attribute this to the absence of distant labels or strong assumptions to group the naturally occurring text by its logical structure.

2.2 Logical Reasoning

Logical reasoning has attracted increasing research attentions recently. Devising specific model architectures and integrating symbolic logic have been proved to be two effective solutions. For example, Huang et al. (2021) and Ouyang et al. (2021) proposed to extract the basic units for logical reasoning, e.g., the elementary discourse or fact units, and then employed graph neural networks (GNNs) to model possible relationships. The graph structure of constituents and predicates can be viewed as a form of prior knowledge pertaining to logical relations. Differently, Betz (2020) generated a synthetic corpus through verbalizing the syllogistic argument schemes with domain-specific predicates, names and templates. Although the synthetic dataset explicitly describes the logical rules, the model pre-trained with it does not achieve better performance on LogiQA than that without pre-training, which is possibly because there is still a gap between the synthetic text and natural language. Besides, Wang et al. (2021) developed a context extension and data augmentation framework based on the extracted logical expressions. Superior performance over its contenders can be observed on the ReClor dataset.

In this paper, we propose a self-supervised contrastive learning approach to enhance the logical reasoning ability of neural models. Orthogonal to existing methods, our approach is endowed with two intriguing merits: 1) it shows strong advantage in utilizing the unlabeled text data, and 2) the symbolic logic is seamlessly introduced into neural models via the guidance of meta-path for automatic data construction.

3 Preliminary

3.1 Contrastive Learning

Contrastive Learning (CL) aims to learn recognizable representations by pulling the semantically similar examples close and pushing apart the dissimilar ones (Hadsell et al., 2006). Given an instance x, a semantically similar example x^+, and a set of dissimilar examples $\mathcal{X}^-$ to x, the objective of CL can be formulated as:

$$\begin{aligned}\mathcal{L}_{\text{CL}} &= L(x, x^+, \mathcal{X}^-) \\ &= -\log \frac{\exp f(x, x^+)}{\sum_{x' \in \mathcal{X}^- \cup \{x^+\}} \exp f(x, x')}\end{aligned} \tag{1}$$

where f is the model to be optimized.

3.2 Meta-Path

Given an arbitrary graph, a meta-path is defined as a sequence of graph nodes connected by the corresponding edges. It can be viewed as a particular data structure expressing the relation between two indirectly connected entities in a knowledge graph (Zeng et al., 2020a; Xu et al., 2021a). Specifically, given a knowledge graph $\mathcal{K}$, where the nodes refer to entities and edges are annotated relations, a meta-path connecting two target entities $\langle e_i, e_j \rangle$ can be given as,

$$e_i \xrightarrow{r_{i,i+1}} e_{i+1} \xrightarrow{r_{i+1,i+2}} \cdots e_{j-1} \xrightarrow{r_{j-1,j}} e_j, \tag{2}$$

where $r_{i,j}$ denotes the relation between entities e_i and e_j. And the set of entities included in the meta-path is denoted as $\mathcal{P}$.

4 Methodology

In this paper, we study the problem of logical reasoning on the task of multiple choice question answering (MCQA). Specifically, given a passage P, a question Q and a set of K options $\mathcal{O} = \{O_1, \cdots, O_K\}$, the goal is to select the correct option O_y, where $y \in [1, K]$. Notably, to tackle this task, we devise a novel pre-training method equipped with contrastive learning, where the abundant knowledge contained in the large-scale Wikipedia documents is firstly explored. We then transfer the learned knowledge to the downstream logical reasoning task.

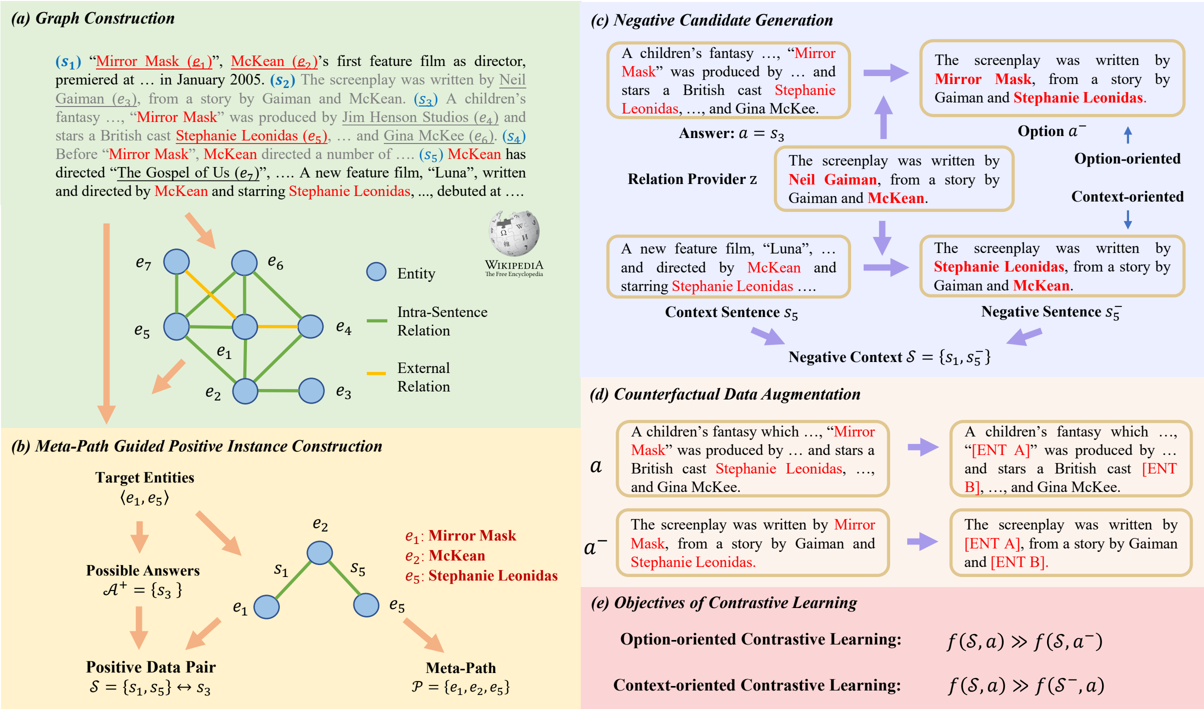

Figure 2: The overall framework of our proposed method. (a) A document $\mathcal{D}$ from Wikipedia and the corresponding built entity-level graph. The sentences in black will extracted as context for pre-training in procedure (b). (b) Given two target entities $\langle e_1, e_5 \rangle$, the possible answers $\mathcal{A}^+$ and meta-path are firstly extracted. The entities involved in the meta-path are denoted as $\mathcal{P}$. And the sentences $\mathcal{S}$ serving as the implicit connecting the entities in $\mathcal{P}$ are extracted as the context for pre-training. A positive instance pair can thus be $(\mathcal{S}, s_3)$, where $\mathcal{S} = \{s_1, s_5\}$. (c) Given a sentence z to provide the alternative relation, the relation modification for negative context sentence and option construction is implemented through entity replacement. The top operation is performed for negative options while the bottom one is to facilitate negative contexts. (d) The counterfactual sentences are generated by entity replacement to eliminate the information shortcut during pre-training. (e) The generated positive and negative samples are used for contrastive learning.

4.1 Symbolic Logical Reasoning

In a sense, in MCQA for logical reasoning, both the given context (i.e., passage and question) and options express certain relations between different logical variables (Figure 1). Go a step further, the relations connecting logical variables contained in the correct option should be able to be inferred from the given context, while that in the wrong options should not. In other words, the context is logically consistent with only the correct option.

In light of this, the training instances for our contrastive learning based pre-training should be in the form of a context-option pair, where the context consists of multiple sentences and expresses the relations between the included constituents, while the option should illustrate the potential relations between parts of the constituents. Nevertheless, it is non-trivial to derive such instance pairs from large-scale unlabeled corpus like Wikipedia documents. Fortunately, according to Xu et al. (2021b), the recognition process for document-level relation extraction, another typical task in demand of logical reasoning, can be formulated as follows,

$$(e_i \xrightarrow{r_{i,i+1}} e_{i+1} \cdots \xrightarrow{r_{j-1,j}} e_j) \rightarrow \langle e_i, r_{i,j}, e_j \rangle, \quad (3)$$

where the left side is equivalent to a meta-path linking entities e_i and e_j comprising a few relation triplets, which can be extracted from the given document; while the right side refers to the implicitly inferred information triplet $\langle e_i, r_{i,j}, e_j \rangle$.

Inspired by this, given an arbitrary document $\mathcal{D} = \{s_1, \cdots, s_m\}$, where s_i is the i-th sentence, we can first build an entity-level graph, denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of entities contained in $\mathcal{D}$ and $\mathcal{E}$ is the set of relations between entities. Notably, to comprehensively capture the relations among entities, we take into account both the external relation from the knowledge graph and the intra-sentence relation. As illustrated in Figure 2 (a), there will be an intra-sentence relation between two entities if they are mentioned by a common sentence. Thereafter, we can derive the pre-training instance pairs according to the meta-paths extracted from the graph, which will be detailed in the following subsections.

4.2 Meta-Path Guided Positive Instance Construction

In fact, in the positive instances, the answer should be logically consistent with the given context. As clarified in Equation 3, it should contain a relation triplet that can be derived based on the given context by logical reasoning. Since we take the intra-sentence relationship into consideration, given a pair of entities contained in the document, we first collect the sentences mentioning both of them as the set of answer candidates. Accordingly, we then try to find a meta-path connecting the entity pair and hence derive the corresponding logically consistent context.

In particular, as shown in Figure 2 (b), given the entity pair $\langle e_i, e_j \rangle$, we denote the collected answer candidates as $\mathcal{A}^+$, and then we use Depth-First Search (Tarjan, 1972) to find a meta-path linking them on $\mathcal{G}$, following Equation 2. Thereafter, the context sentences $\mathcal{S}$ corresponding to the answer candidates in $\mathcal{A}^+$ are derived by retrieving those sentences undertaking the intra-sentence relations during the search algorithm. Finally, for each answer candidate $a \in \mathcal{A}^+$, the pair $(\mathcal{S}, a)$ is treated as a positive context-answer pair to facilitate our contrastive learning. The details of positive instance generation algorithm are described in Appendix A.

4.3 Negative Instance Generation

In order to obtain the negative instances (i.e., negative context-option pairs) where the option is not logically consistent with the context, the most straightforward method is randomly sampling the sentences from different documents. However, the approach could lead to trivial solution by simply checking whether the entities involved in each option are the same as those in the given context. In the light of this, we resort to directly breaking the logical consistency of the positive instance pair by modifying the relation rather than the entities in the context or the option, to derive the negative instance pair.

In particular, given a positive instance pair $(\mathcal{S}, a)$, we devise two negative instance generation methods: the context-oriented method and the option-oriented method, which focus on generating negative pairs by modifying the relations involved in the context $\mathcal{S}$ and answer a of the positive pair, respectively. Considering that the relation is difficult to be extracted, especially, the intra-sentence relation, we propose to implement this reversely via the entity replacement. In particular, for the option-oriented method, suppose that $\langle e_i, e_j \rangle$ is the target entity pair for retrieving the answer a, we first randomly sample a sentence z that contains at least a different entity pair $\langle e_a, e_b \rangle$ from $\langle e_i, e_j \rangle$ as the relation provider. We then obtain the negative option by replacing the entities e_a and e_b in z with e_i and e_j, respectively. The operation is equivalent to replacing the relation contained in a with that in z. Formally, we denote the operation as

$$a^- = \text{Relation\_Replace}(z \rightarrow a).$$

Pertaining to the context-oriented negative instance generation method, we first randomly sample a sentence $s_i \in \mathcal{S}$, and then conduct the modification process as follows,

$$s_i^- = \text{Relation\_Replace}(z \rightarrow s_i),$$

where the entity pair to be replaced in s_i should be contained in the meta-path corresponding to the target entity pair $\langle e_i, e_j \rangle$. Accordingly, the negative context can be written as $\mathcal{S}^- = \mathcal{S} \setminus \{s_i\} \cup \{s_i^-\}$. Figure 2 (c) illustrates the above operations on both the answer and context sentence.

4.4 Counterfactual Data Augmentation

According to (Ko et al., 2020; Lai et al., 2021), the neural models are adept in finding a trivial solution through the illusory statistical information in datasets to make correct predictions, which often leads to inferior generalization. In fact, this issue can also occur in our scenario. In particular, since the correct answer is a real sentence and describes a real world fact, while the negative option is synthesized by entity replacement, which may conflict with the commonsense knowledge, the pre-trained language model tends to identify the correct option directly by judging its factuality rather than its logical consistency with the given context. For example, as shown in Figure 2 (d) (left), the answer a is a fact while the synthetic negative option a^- is not. As a result, the language model will deem a as correct, simply due to that the other option a^- conflicts with the world knowledge.

To overcome this problem, we develop a simple yet effective counterfactual data augmentation method to further improve the capability of logical reasoning (Zeng et al., 2020b). Specifically, given the entities $\mathcal{P}$ that are involved in the meta-path, we randomly select some entities from $\mathcal{P}$ and replace their occurrences in the context and the answer of the positive instance pair $(\mathcal{S}, a)$ with the entities extracted from other documents. In this manner, the positive instance also contradicts to the world knowledge. Notably, considering that the positive and negative instance pairs should keep the same set of entities, we also conduct the same replacement for a^- or $\mathcal{S}^-$, if they mention the selected entities. As illustrated in Figure 2 (d) (right), a counterfactual instance can be generated by replacing *Mirror Mask* and *Stephanie Leonidas* in a and a^- with **[ENT A]** and **[ENT B]**, where **[ENT A]** and **[ENT B]** are arbitrary entities, and that for $\mathcal{S}$ and $\mathcal{S}^-$ are provided in Appendix D. Ultimately, the key to infer the correct answer lies in the accurate inference of the logical relation between entities **[ENT A]** and **[ENT B]** implied in each context-option pair.

4.5 Contrastive Learning based Pre-training

Since we have two ways for negative instance generation, accordingly, we have two contrastive learning schemes: option-oriented CL and context-oriented CL. Let $\mathcal{A}^-$ be the set of all constructed negative options with respect to the correct option a. The option-oriented CL can be formulated as:

$$\mathcal{L}_{\text{OCL}} = L(\mathcal{S}, a, \mathcal{A}^-). \tag{4}$$

In addition, given $\mathcal{C}^-$ as the set of all generated negative contexts corresponding to $\mathcal{S}$, the objective of context-oriented CL can be written as:

$$\mathcal{L}_{\text{CCL}} = L(a, \mathcal{S}, \mathcal{C}^-). \tag{5}$$

To avoid the catastrophic forgetting problem, we also add the MLM objective during pre-training and the final loss is given as:

$$\mathcal{L} = \mathcal{L}_{\text{OCL}} + \mathcal{L}_{\text{CCL}} + \mathcal{L}_{\text{MLM}}. \tag{6}$$

4.6 Fine-tuning

In the fine-tuning stage, to solve the task of MCQA, we adopt the following loss function:

$$\mathcal{L}_{\text{QA}} = -\log \frac{\exp f(P, Q, O_y)}{\sum_i \exp f(P, Q, O_i)}, \tag{7}$$

where O_y is the ground-truth option for the question Q, given the passage P. As for the model architecture, we employ a 2-layer MLP to output a score for each triplet of inputs.

Prompt Tuning Although our proposed pre-training method is task-oriented and its training instance and objective are both developed towards MCQA, there still exists a gap between the input of the pre-training and fine-tuning. As aforementioned, the input of our pre-training stage comprises a context and a option, while that of the fine-tune stage additionally has the question as a condition. To overcome this, we propose to employ the prompt tuning (Lester et al., 2021) technique, which is designed to fully exploit the knowledge obtained during large-scale pre-training by converting the fine-tuning task into the same format of pre-training.

Considering that the objectives in the two stages in our method are essentially the same, following (Lester et al., 2021), we simply append several unseen tokens as a prefix of the question and option pair, and randomly initialize their embedding. The ultimate input sequence for fine-tuning is $\langle P, [\text{prefix}], Q, O_i \rangle$. Meanwhile, the parameters of output layer, i.e., a 2-layer MLP, are also initialized with the pre-trained weights, rather than being trained from scratch.

5 Experiment

5.1 Dataset

ReClor (Yu et al., 2020) is extracted from logical reasoning questions of standardized graduate admission examinations. The held-out test set is further divided into EASY and HARD subsets, denoted as test-E and test-H, respectively. The instances in test-E are biased and can be solved even without knowing contexts and questions by neural models. A leader board[3] is also host for public evaluation.

LogiQA (Liu et al., 2020) consists of 8,678 multiple-choice questions collected from National Civil Servants Examinations of China and are manually translated into English by experts. The dataset is randomly split into train/dev/test sets with 7,376/651/651 samples, respectively. LogiQA contains various logical reasoning types, e.g., categorical reasoning and sufficient conditional reasoning.

5.2 Baseline

DAGN (Huang et al., 2021) is a discourse-aware graph network that reasons on the discourse structure of texts. It is based on elementary discourse

[3] https://bit.ly/3CHTwNl.

Model / Dataset	ReClor				LogiQA	
	Dev	Test	Test-E	Test-H	Dev	Test
XLNet	62.0	56.0	75.7	40.5	—	—
RoBERTa	62.6	55.6	75.5	40.0	35.0	35.3
DAGN	65.2	58.2	76.1	44.1	35.5	38.7
DAGN (Aug)	65.8	58.3	75.9	44.5	36.9	39.3
LReasoner (RoBERTa)‡	64.7	58.3	77.6	43.1	—	—
Focal Reasoner	66.8	58.9	77.1	44.6	**41.0**	40.3
MERIt	66.8	59.6	78.1	45.2	40.0	38.9
MERIt + LReasoner	67.4	60.4	78.5	46.2	—	—
MERIt + Prompt	**69.4**	**61.6**	**79.3**	**47.8**	39.9	**40.7**
MERIt + Prompt + LReasoner	67.3	61.4	79.8	46.9	—	—
ALBERT	69.1	66.5	76.7	58.4	38.9	37.6
MERIt (AlBERT)	74.2	70.1	81.6	61.0	43.7	**42.5**
MERIt (AlBERT) + Prompt	**74.7**	**70.5**	**82.5**	**61.1**	**46.1**	41.7

Table 1: The overall results on ReClor and LogiQA. We adopt the **accuracy** as the evaluation metric and all the baselines are based on RoBERTa except specific statement. For each model we repeated training for 5 times using different random seeds and reported the average results. ‡: The results are reproduced by ourselves.

units and discourse relations. DAGN (Aug) is a variant that augments the graph features.

Focal Reasoner (Ouyang et al., 2021) is a fact-driven logical reasoning model, which builds super-graphs on the top of fact units as the basis for logical reasoning.

LReasoner (Wang et al., 2021) includes a context extension framework and a data augmentation algorithm, which are all conducted based on the extracted logical expressions. This method has achieved new state-of-the-art performance on ReClor recently.

Besides, we also compare the performance with the directly fine-tuned large pre-trained language models, including RoBERTa, XLNet and ALBERT.

5.3 Implementation Detail

We further pre-trained RoBERTa and ALBERT on Wikipedia for another 500 and 100 steps, respectively, and the batch size for pre-training is set to 4,096. All experiments conducted on downstream tasks are repeated for 5 times with different random seeds. Due to the space limit, More implementation details can be found in Appendix C.

6 Result and Analysis

6.1 Overall Results

The overall results on ReClor and LogiQA are shown in Table 1. It can be observed that 1) MERIt outperforms all the strong baselines using the same backbone with significant improvements. Besides, our method achieves the new state-of-the-art performance on both datasets. 2) The prompt-tuning further enhances our model with a significant performance margin. This may suggest that the gap between our pre-training method and fine-tuning tasks is relatively small. 3) MERIt achieves better performance on the more difficult split of ReClor (Test-H), indicating that our pre-training method is less affected by the statistical short-cut (Yu et al., 2020). 4) MERIt + Prompt does not benefit from the framework of LReasoner significantly. This is probably because the basic knowledge about logic rules has been covered in our method. Besides, since Wang et al. (2021) merely reported their best results on the test set, we also report those on both datasets in Appendix B.

6.2 Ablation Study

Table 2 shows the results of our ablation studies. To observe the impacts brought by the meta-path strategy, we built a baseline model without the meta-path strategy by randomly selecting the sentences in a passage to form the context-answer pairs.

From this table we can conclude that: 1) the model without counterfactual data augmentation (- DA) has a severe performance degradation. It suggests that the counterfactual data is essential for MERIt to conduct logical reasoning. As for the ratio of original data to the counterfactual one, on test set, we found that 1:3 (+ DA^3) leads to better performance using prompt tuning while 1:2 (+ DA^2) obtains the best performance using simple

Model	Dev	Dev (P.)	Test	Test (P.)
MERIt	66.8	69.4	59.6	61.6
- DA	63.0	64.5	57.9	59.8
+ DA^2	65.3	67.8	**60.2**	61.3
+ DA^3	66.2	68.0	59.3	**61.9**
- Option-oriented CL	63.8	65.4	58.9	61.5
- Context-oriented CL	64.0	66.5	58.8	60.2
- Meta-Path	64.8	65.1	58.0	60.8

Table 2: Performance comparisons on ReClor between different variants of MERIt. *DA* means data augmentation and *DA*^{N} refers to 1:N ratio of the original data to the augmented data. *P.* is short for *Prompt Tuning*.

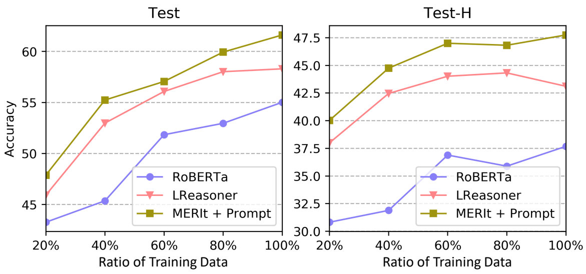

Figure 3: Results on the test set (left) and the test-H set (right) of ReClor.

fine-tuning. 2) The model without the guidance of meta-path (- Meta-Path) demonstrates a much worse performance than MERIt, indicating that the meta-path strategy plays an important role by discovering the potential logic structure. 3) Considering the results of models without the objectives of option-oriented CL and context-oriented CL, it can be seen that both contrastive learning schemes are beneficial for logical reasoning. In addition, the context-oriented CL is more effective than option-oriented CL. One possible reason to this is that the context-oriented CL is more diverse in format since each sentence can be disturbed while the option-oriented CL will make the model pay more attention to the option, leading to a worse generalization during fine-tuning.

6.3 Performance with Limited Training Data

Figure 3 shows the accuracy on the test set and test-H set of ReClor with respect to different amount of training data. We reported the average results of MERIt + Prompt, LReasoenr and RoBERTa. It can be observed that: 1) With the scale of training data becoming larger, the performance of all models achieves improvements. 2) MERIt + Prompt shows better performance under low resource, especially on test-H. Our method trained on 40% data has achieved comparable performance with RoBERTa. In addition, on test-H, our method outperforms

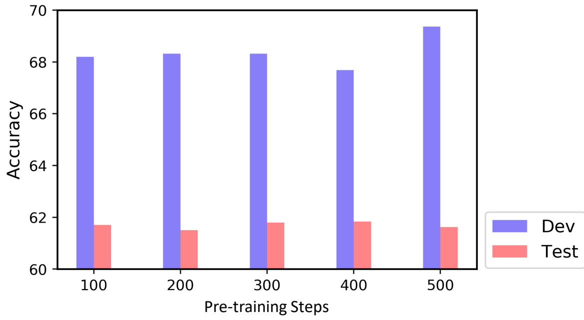

Figure 4: The prompt-tuning results on ReClor using the models pre-trained with different steps.

RoBERTa and LReasoner trained on full dataset using only 20% and 40% training data, respectively, evidently demonstrating the generalization capability of our method. 3) Further improvements to LReasoner become insignificant when consuming more training data. This suggests that the basic logic rules can be easily fitted.

6.4 Effect of Pre-training Steps

In order to explore the effects of pre-training steps, we fine-tuned the models pre-trained for different steps on ReClor and the results are shown in Figure 4. From the histogram we can find that our method achieves the best performance on dev set at 500 steps. Besides, the model pre-trained with 100 steps (using only around 410k samples) has achieved comparable performance with the best one, indicating that our method is very competitive with few training iterations.

7 Conclusion and Future Work

In this paper, we present MERIt, a meta-path guided contrastive learning method to facilitate logical reasoning via self-supervised pre-training. MERIt is built upon the meta-path strategy for automatic data construction and the counterfactual data augmentation to eliminate the information shortcut during pre-training. With the evaluation on two logical reasoning benchmarks, our method has obtained significant improvements over strong baselines relying on task-specific model architecture or augmentation of original dataset. Pertaining to the further work, we plan to strengthen our method from both data construction and model architecture design angles. More challenging instances are expected to be constructed if multiple meta-paths can be considered at the same time. Besides, leveraging GNNs may bring better interpretability and generalization since the graph structure can be integrated into both pre-training and fine-tuning stages.

References

Gregor Betz. 2020. Critical thinking for language models. *CoRR*, abs/2009.07185.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *NeurIPS*.

Xiang Deng, Yu Su, Alyssa Lees, You Wu, Cong Yu, and Huan Sun. 2021. ReasonBERT: Pre-trained to reason with distant supervision. In *EMNLP*. ACL.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186. ACL.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *EMNLP*. ACL.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: retrieval-augmented language model pre-training. *CoRR*, abs/2002.08909.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *CVPR*, pages 1735–1742. IEEE.

Yinya Huang, Meng Fang, Yu Cao, Liwei Wang, and Xiaodan Liang. 2021. DAGN: discourse-aware graph network for logical reasoning. In *NAACL-HLT*, pages 5848–5855. ACL.

Miyoung Ko, Jinhyuk Lee, Hyunjae Kim, Gangwoo Kim, and Jaewoo Kang. 2020. Look at the first sentence: Position bias in question answering. In *EMNLP*, pages 1109–1121. ACL.

Yuxuan Lai, Chen Zhang, Yansong Feng, Quzhe Huang, and Dongyan Zhao. 2021. Why machine reading comprehension models learn shortcuts? In *Findings of ACL/IJCNLP*, pages 989–1002. ACL.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *ICLR*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *EMNLP*. ACL.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. LogiQA: A challenge dataset for machine reading comprehension with logical reasoning. In *IJCAI*, pages 3622–3628.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Siru Ouyang, Zhuosheng Zhang, and Hai Zhao. 2021. Fact-driven logical reasoning. *CoRR*, abs/2105.10334.

Yujia Qin, Yankai Lin, Ryuichi Takanobu, Zhiyuan Liu, Peng Li, Heng Ji, Minlie Huang, Maosong Sun, and Jie Zhou. 2021. ERICA: improving entity and relation understanding for pre-trained language models via contrastive learning. In *ACL/IJCNLP*, pages 3350–3363. ACL.

Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. 2020. GCC: graph contrastive coding for graph neural network pre-training. In *KDD*, pages 1150–1160. ACM.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Robert Endre Tarjan. 1972. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160.

Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. 2006. Fast random walk with restart and its applications. In *ICDM*, pages 613–622. IEEE.

Siyuan Wang, Wanjun Zhong, Duyu Tang, Zhongyu Wei, Zhihao Fan, Daxin Jiang, Ming Zhou, and Nan Duan. 2021. Logic-driven context extension and data augmentation for logical reasoning of text. In *NAACL-HLT*, pages 5642–5650. ACL.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *EMNLP: System Demonstrations*, pages 38–45. ACL.

Wang Xu, Kehai Chen, and Tiejun Zhao. 2021a. Discriminative reasoning for document-level relation extraction. In *Findings of ACL/IJCNLP*, pages 1653–1663. ACL.

Wang Xu, Kehai Chen, and Tiejun Zhao. 2021b. Document-level relation extraction with reconstruction. In *AAAI*, pages 14167–14175.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*, pages 5754–5764.

Yang You, Jing Li, Sashank J. Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. 2020. Large batch optimization for deep learning: Training BERT in 76 minutes. In *ICLR*.

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. 2020. Reclor: A reading comprehension dataset requiring logical reasoning. In *ICLR*.

Shuang Zeng, Yuting Wu, and Baobao Chang. 2021. SIRE: separate intra- and inter-sentential reasoning for document-level relation extraction. In *Findings of ACL/IJCNLP*, pages 524–534. ACL.

Shuang Zeng, Runxin Xu, Baobao Chang, and Lei Li. 2020a. Double graph based reasoning for document-level relation extraction. In *EMNLP*, pages 1630–1640. ACL.

Xiangji Zeng, Yunliang Li, Yuchen Zhai, and Yin Zhang. 2020b. Counterfactual generator: A weakly-supervised method for named entity recognition. In *EMNLP*, pages 7270–7280. ACL.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In *ICML*, pages 11328–11339. PMLR.

A DFS-based Algorithm for Meta-Path Extraction

Algorithm 1 The DFS algorithm to obtain the meta-paths.

Input: The graph $\mathcal{G} = (\mathcal{E}, \mathcal{V})$; The sentences of the document $\mathcal{D} = \{s_1, \cdots, s_m\}$; The entity set of the i-th sentence $\mathcal{V}_i$;
Output: $\mathcal{P}$, $\mathcal{S}$, and $\mathcal{A}^+$;

for each $(e_i, e_j) \in \mathcal{V} \times \mathcal{V}$ and $i \neq j$ **do**
  $\mathcal{A}^+ = \{s_k | e_i \in \mathcal{V}_k, e_j \in \mathcal{V}_k\}$;
  $\mathcal{D}' = \mathcal{D} \setminus \mathcal{A}^+$;
  $\mathrm{cond}, \mathcal{P}, \mathcal{S} \leftarrow$ DFS($e_i, \{e_i\}, \varnothing, e_j, \mathcal{G}, \mathcal{D}'$);
  if cond is TRUE and $\mathcal{A}^+$ is not $\varnothing$ **then**
    return $\mathcal{A}^+, \mathcal{P},\ \mathcal{S}$;
  end if
end for
return $\varnothing, \varnothing, \varnothing$;

function DFS($e_i, \mathcal{P}', \mathcal{S}', e_d, \mathcal{G} = (\mathcal{E}, \mathcal{V}), \mathcal{D}'$)
  if $e_i = e_d$ **then**
    return TRUE, $\mathcal{P}', \mathcal{S}'$;
  end if
  for each $(e_j, s_k) \in \mathcal{V} \times \mathcal{D}'$ and $(e_i, e_j) \in \mathcal{E}, e_j \in \mathcal{V}_k$ **do**
    $\mathcal{G}' = (\mathcal{E}, \mathcal{V} \setminus \{e_j\})$;
    $\mathcal{P}'' = \mathcal{P}' \cup \{e_j\}$;
    if $e_i \in \mathcal{V}_k$ **then**
      $\mathcal{D}'' = \mathcal{D}' \setminus \{s_k\}$;
      $\mathcal{S}'' = \mathcal{S}' \cup \{s_k\}$;
    else
      $\mathcal{D}'' = \mathcal{D}', \mathcal{S}'' = \mathcal{S}'$;
    end if
    return DFS($e_j, \mathcal{P}'', \mathcal{S}'', e_d, \mathcal{G}', \mathcal{D}''$);
  end for
  return FALSE, $\varnothing$, $\varnothing$;
end function

B Best Performance on Test Set

The best results on the test sets of both datasets are reported in Table 3. As shwon in the table, our method also outperforms the previous SOTA baseline, i.e., LReasoner, with significant improvements.

C Implementation Detail

C.1 Data Construction

During the data construction process, we have employed two tricks to improve the complexity of the pretext task:

1. For the sentence z as the relation provider for negative instance construction, the sentences from the document are primarily to be considered because they share the same entities with the context or describe the same topic. This can also be viewed as a trick to avoid trivial solution by checking whether the samples come from the same domain. Another problem is that if z comes from the same document, taking the option-oriented method as example, the replacement may not work if $e_i = e_a$ and $e_j = e_b$. To address it, we will change the order of the entities to be replaced, i.e., swapping the mentions of e_i and e_j.

2. Similarly, for counterfactual data augmentation, supposing the extracted meth-path of a training instance connects an entity pair $\langle\, e_i, e_j \,\rangle$, e_i and e_j are always considered to be replaced for generating counterfactual data. And thus the sets of answer candidates $\mathcal{A}^+$ constructed from other documents, where the corresponding meta-paths also link $\langle\, e_i, e_j \,\rangle$, can be employed as negative candidates directly. The motivation of the trick is to avoid modifications on the original texts as many as possible.

C.2 Pre-training Setting

We employed the model implementation of Transformer from Huggingface (Wolf et al., 2020) and pytorch[4] framework. The corpus for pre-training is generated from the dataset provided by Qin et al. (2021)[5], which includes the pre-processed passages from Wikipedia and the recognized entities with their distantly annotated relations. The generated corpus contains one million samples and each sample has 3 negative options.

During pre-training, we adopted the LAMB (You et al., 2020) optimizer, warming up the learning rate to the peak and then linearly decaying it. It takes 32 hours on 4 RTX 2080Ti GPUs for RoBERTa pre-training and 3 days on 2 TeslaT4 GPUs for ALBERT pre-training. Other hyper-parameters for pre-training are reported in Table 4.

C.3 Hyper-parameters for Fine-tuning

The random seeds we utilized for repeated experiments are 42, 43, 44, 45 and 4321. The hyper-parameters for fine-tuning are shown in Table 6.

[4]https://pytorch.org.
[5]https://github.com/thunlp/ERICA.

Model / Dataset	ReClor				LogiQA	
	Dev	Test	Test-E	Test-H	Dev	Test
LReasoner (RoBERTa)	66.2	62.4	81.4	47.5	38.1	40.6
MERIt	67.8	60.7	79.6	45.9	**42.4**	41.5
MERIt + Prompt	**70.2**	**62.6**	80.5	**48.5**	39.5	**42.4**
LReasoner (ALBERT)	73.2	70.7	81.1	62.5	41.6	41.2
MERIt (ALBERT)	73.2	71.1	83.6	61.3	43.9	**45.3**
MERIt (ALBERT) + Prompt	**75.0**	**72.2**	**82.5**	**64.1**	**45.8**	43.8

Table 3: The best results on test set of ReClor and LogiQA.

	ALBERT	RoBERTa
Batch Size	4096	4096
Peak Learning Rate	5e-5	1e-4
Training Steps	100	500
Warmup Proportion	0.2	0.1
Weight Decay	0.01	0.01
Adam ϵ	1e-6	1e-6
Adam β_1	0.9	0.9
Adam β_2	0.98	0.98
Max Sequence Length	256	320
Gradient Clipping	5.0	5.0
Hidden Size of MLP	8192	2048

Table 4: Hyper-parameters for ALBERT and RoBERTa during pre-training, respectively.

Model	Dev	Test	Test-E	Test-H
RoBERTa	35.8	35.7	44.5	28.8
MERIt (500 steps)	**39.0**	35.2	41.8	30.0
100 steps	37.5	**38.1**	**47.5**	30.6
200 steps	38.1	38.0	47.3	30.7
300 steps	37.4	36.4	43.6	30.7
400 steps	38.5	35.9	42.5	30.7
ALBERT	43.6	40.2	46.6	35.2
MERIt (ALBERT)	**46.3**	**44.6**	**51.8**	**38.9**

Table 5: Results of Linear Probing on ReClor.

D Case Study for Generated Examples

Figure 5 shows the constructed examples for contrastive learning as well as the corresponding counterfactual examples.

E Results for Linear Probing

Table 5 shows the results of linear probing on ReClor, where we used a single linear layer as the output layer and only fine-tuned its parameters. As shown in the table, MERIt (100 steps) and MERIt (ALBERT) outperform RoBERTa and ALBERT on both dev and test set, respectively.

F A Different View from Contrastive Graph Representation Learning

To understand why the pre-training approach can promote logical reasoning, we provide a different view from the contrastive learning for graphs. Following Qiu et al. (2020), x and x^+ in Equation 1 are different sub-graphs extracted from the same graph through random walk with restart (Tong et al., 2006) while x^- is sub-graph sampled from a different graph. To avoid the trivial solution by simply checking whether the node indices of two sub-graphs match, they also developed an anonymization operation by relabeling the nodes of each sub-graph. In fact, our proposed method can be taken as a special case of graph contrastive learning. Firstly, the context and answer based on the meta-path can be viewed as sub-graphs of $\mathcal{G}$. In particular, the answer is the sub-graph with only two nodes (the two entities connected by the meta-path). Secondly, the entity replacement for negative candidates construction and counterfactual data generation play similar roles with the anonymization operation. Both of them aim at guiding the model focus on the logical/graph structure. The only assumption our approach built upon is that inferring the consistency defined in Equation 3 is in demand of logical reasoning, which has already been explored in many studies for document-level relation extraction (Zeng et al., 2021, 2020a).

	ALBERT		RoBERTa	
	ReClor	LogiQA	ReClor	LogiQA
Batch Size	24	24	24	16
Peak Learning Rate	2e-5♣/3e-5	2e-5	1e-5♣/1.5e-5♠	8e-6
Epoch	10	10	10	10
Warmup Proportion	0.1	0.1	0.1	0.2
Weight Decay	0.01	0.01	0.01	0.01
Adam ϵ	1e-6	1e-6	1e-6	1e-6
Adam β_1	0.9	0.9	0.9	0.9
Adam β_2	0.98	0.98	0.98	0.98
Max Sequence Length	256♣/231♠	256♣/231♠	256♣/231♠	256♣/231♠
Prefix Length	0♣/5♠	0♣/5♠	0♣/5♠	0♣/5♠
Gradient Clipping	0.0	0.0	0.0	0.0
Dropout	0.1	0.0♣/0.1♠	0.1	0.1

Table 6: Hyper-parameters for fine-tuning on ReClor and LogiQA. ♣: Fine-Tuning. ♠: Prompt Tuning.

Example 1 (Option-oriented CL)
Context:
Napoleon appointed his brother Louis Bonaparte to the Kingdom of Holland in May 1806. The Dutch rebellion first broke out in Amsterdam on 14–15 November.
Negative Candidates:
- Since their trade was badly damaged by Napoleon's Continental System, the French people were ready to throw off the Dutch yoke.
- However, on 9 July 1810, the French emperor extinguished the kingdom and annexed the Dutch to the Napoleon.
- Depressed by the loss of his son in Napoleon, the French civil leader Dutch responded ineffectively to the crisis.

Answer:
The Dutch contributed only 17,300 soldiers to Napoleon's armies in 1811–1813, but their severe casualties in the French invasion of Russia shocked the population.

A Counterfactual Sample of Example 1
Context:
The Din rebellion first broke out in Amsterdam on 14–15 November. Bihar appointed his brother Louis Bonaparte to the Kingdom of Holland in May 1806 .
Negative Candidates:
- Since their trade was badly damaged by French's Continental System , the Din people were ready to throw off the Bihar yoke .
- In early November, Din corps commander Ferdinand von Wintzingerode sent a 3,500-man "Streifkorps" led by Alexander Khristoforovich Benckendorff into Bihar.
- In early November, Bihar corps commander Ferdinand von Wintzingerode sent a 3,500-man "Streifkorps" led by Alexander Khristoforovich Benckendorff into Din.

Answer:
The Dutch contributed only 17,300 soldiers to Napoleon's armies in 1811–1813, but their severe casualties in the French invasion of Russia shocked the population.

Example 2 (Context-oriented CL)
Context:
Napoleon appointed his brother Louis Bonaparte to the Kingdom of Holland in May 1806. The Dutch rebellion first broke out in Amsterdam on 14–15 November.
Negative Contexts:
- Depressed by the loss of his son in Napoleon, the French civil leader Kingdom of Holland responded ineffectively to the crisis. The Dutch rebellion first broke out in Amsterdam on 14–15 November.
- Since their trade was badly damaged by Kingdom of Holland's Napoleon, the Dutch people were ready to throw off the French yoke. The Dutch rebellion first broke out in Amsterdam on 14–15 November.
- Depressed by the loss of his son in Russia, the Napoleon civil leader Kingdom of Holland responded ineffectively to the crisis. The Dutch rebellion first broke out in Amsterdam on 14–15 November ..

Answer:
The Dutch contributed only 17,300 soldiers to Napoleon's armies in 1811–1813, but their severe casualties in the French invasion of Russia shocked the population.

A Counterfactual Sample of Example 2
Context:
Bihar appointed his brother Louis Bonaparte to the Kingdom of Holland in May 1806. The Din rebellion first broke out in Amsterdam on 14–15 November.
Negative Contexts:
- The Din rebellion first broke out in Amsterdam on 14–15 November. Since their trade was badly damaged by Kingdom of Holland's Continental System, the Din people were ready to throw off the Bihar yoke.
- The Din rebellion first broke out in Amsterdam on 14–15 November. Depressed by the loss of his son in Kingdom of Holland, the French civil leader Bihar responded ineffectively to the crisis.
- Since their trade was badly damaged by Bihar's Continental System , the Kingdom of Holland people were ready to throw off the French yoke. The Din rebellion first broke out in Amsterdam on 14–15 November.

Answer:
The Din contributed only 17,300 soldiers to Bihar's armies in 1811–1813, but their severe casualties in the French invasion of Russia shocked the population.

Figure 5: Two cases of the generated and the counterfactual examples. The target entities used for extracting meta-path are colored in red.